

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-122901

(43)Date of publication of application : 28.04.2000

(51)Int.Cl.

G06F 11/34

(21)Application number : 10-289661

(71)Applicant : HITACHI LTD

(22)Date of filing : 12.10.1998

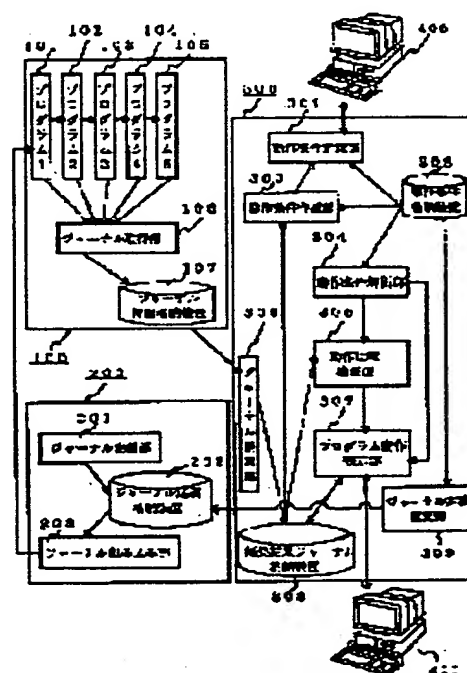
(72)Inventor : USAMI MASAYUKI
NAKANO TOSHIHIKO

(54) JOURNAL ACQUIREMENT ANALYSIS DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a journal acquirement analysis device which can easily set a journal and can easily compare/analyze acquired journal information and design data.

SOLUTION: A definition part 201 defines which starting cause in a processing unit in a program is set to be a journal point. A journal incorporation part 203 detects the journal point from the program and incorporates the journal. An operation condition as design data is defined by a definition part 301. An operation comparison certification part 305 compares acquired journal information with the operation condition and displays the result to a terminal equipment 401.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-122901
(P2000-122901A)

(43) 公開日 平成12年4月28日 (2000.4.28)

(51) Int.Cl.⁷
G 0 6 F 11/34

識別記号

F I
G 0 6 F 11/34

テマコード* (参考)
A 5 B 0 4 2

審査請求 未請求 請求項の数 6 O L (全 14 頁)

(21) 出願番号 特願平10-289661
(22) 出願日 平成10年10月12日 (1998.10.12)

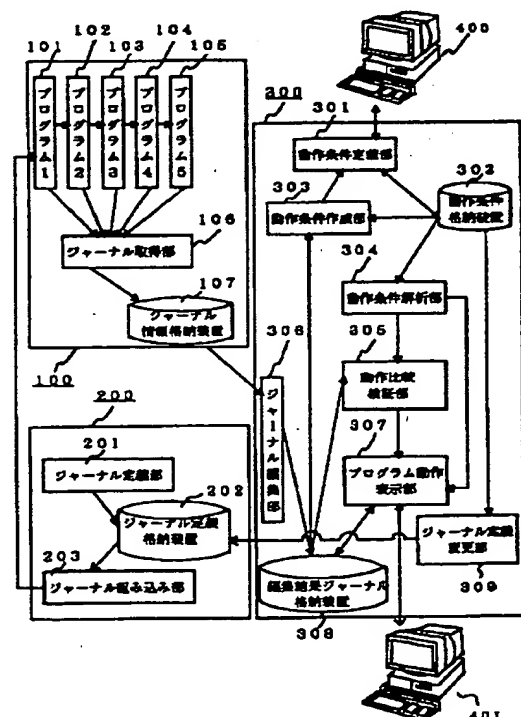
(71) 出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地
(72) 発明者 宇佐美 政行
茨城県日立市大みか町五丁目2番1号 株
式会社日立製作所大みか工場内
(72) 発明者 中野 利彦
茨城県日立市大みか町五丁目2番1号 株
式会社日立製作所大みか工場内
(74) 代理人 100093872
弁理士 高崎 芳祐
Fターム(参考) 5B042 GA10 GB05 GC08 LA02 MA14
MC13 MC35 NN09 NN13 NN14
NN23 NN43

(54) 【発明の名称】 ジャーナル取得解析装置

(57) 【要約】

【課題】 ジャーナル設定が容易で、取得したジャーナル情報と設計データとの比較検討が容易に行えるジャーナル取得解析装置を実現する。

【解決手段】 プログラム中のどの処理単位のどの起動要因をジャーナルポイントとするかを定義部201で定義し、ジャーナル組み込み部203はそのジャーナルポイントをプログラムから検出してジャーナルを組み込む。また設計データとしての動作条件はその定義部301で定義される。動作比較検証部305は、取得したジャーナル情報と動作条件とを比較し、その結果を端末装置401へ表示する。



【特許請求の範囲】

【請求項1】 解析対象プログラムを構成する処理単位のうち、どの処理単位がどの処理要因で起動されたときにその処理単位の始点あるいは始点と終点とにジャーナルポイントを設定するかをユーザが定義して格納しておくためのジャーナル定義格納手段と、

この手段により定義されたプログラムのジャーナルポイントを検出してその位置へジャーナルを組み込むためのジャーナル組み込み手段と、

前記処理単位毎に、その起動要因と起動要因対応の予想動作時間及び次に起動する起動処理単位とその起動要因をユーザが定義し格納するための動作条件定義格納手段と、

解析対象プログラム実行時に前記ジャーナル組み込み手段により組み込まれたジャーナルにより出力されたジャーナル情報を取得し格納するためのジャーナル情報取得格納手段と、

この手段により取得格納されたジャーナル情報と前記動作条件定義格納手段に格納された動作条件とをそれぞれ解析編集して比較し、その比較結果をユーザに表示するための比較表示手段と、

を備えたことを特徴とするジャーナル取得解析装置。

【請求項2】 解析対象プログラムを構成する処理単位の始点と終点をジャーナルポイントとして検出し、その位置へジャーナルを組み込むためのジャーナル組み込み手段と、

前記処理単位毎に、その起動要因と起動要因対応の予想動作時間及び次に起動する起動処理単位とその起動要因をユーザが定義し格納するための動作条件定義格納手段と、

解析対象プログラム実行時に前記ジャーナル組み込み手段により組み込まれたジャーナルにより出力されたジャーナル情報を取得し格納するためのジャーナル情報取得格納手段と、

この手段により取得格納されたジャーナル情報と前記動作条件定義格納手段に格納された動作条件とをそれぞれ解析編集して比較し、その比較結果をユーザに表示するための比較手段と、

を備えたことを特徴とするジャーナル取得解析装置。

【請求項3】 解析対象プログラムを構成する処理単位毎に、その起動要因と起動要因対応の予想動作時間及び次に起動する起動処理単位とその起動要因をユーザが定義し格納するための動作条件定義格納手段と、

この手段により定義された動作条件からジャーナルポイントを自動生成してその位置へジャーナルを組み込むためのジャーナル組み込み手段と、

解析対象プログラム実行時に前記ジャーナル組み込み手段により組み込まれたジャーナルにより出力されたジャーナル情報を取得し格納するためのジャーナル情報取得格納手段と、

この手段により取得格納されたジャーナル情報と前記動作条件定義格納手段に格納された動作条件とをそれぞれ解析編集して比較し、その比較結果をユーザに表示するための比較表示手段と、

を備えたことを特徴とするジャーナル取得解析装置。

【請求項4】 前記比較表示手段は、前記取得されたジャーナル情報と前記定義された動作条件とが一致しない処理単位と一致する処理単位とを区別して図示する機能と、処理単位を示す図形を指示手段で指示したときに当該処理単位に関する前記ジャーナル情報を表示する機能とを備えたことを特徴とする請求項1～3の内の1つに記載のジャーナル取得解析装置。

【請求項5】 前記ジャーナル情報取得格納手段は、解析対象プログラムが複数台の計算機にネットワークを介して分割して搭載されている場合に、前記ジャーナル情報をネットワークを介して1つの計算機に集約して格納する集約手段を有していることを特徴とする請求項1～3の内の1つに記載のジャーナル取得解析装置。

【請求項6】 解析対象プログラムに組み込むジャーナルを複数のグループに分割するとともに、前記ジャーナル情報取得手段は、前記ジャーナルのグループ毎に当該ジャーナルにより得られたジャーナル情報を分類して格納することを特徴とする請求項1～3の内の1つに記載のジャーナル取得解析装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、計算機システムにおけるジャーナル取得とそれを解析するためのジャーナル取得解析装置に関するものである。

【0002】

【従来の技術】金融機関等のオンラインシステム、鉄道の運行管理システム等では、システム動作時にジャーナル（システム履歴）情報を取得しておき、障害発生時のシステム回復作業にこの情報が参照・検証される。このようなジャーナル情報は、動作プログラムのデータを取得したいところ（ジャーナルポイント）をプログラマが決め、ここにジャーナル情報を取得するためのコマンド（以下、このコマンドも混乱がない限りジャーナルという）を入れる。そして、得られたジャーナル情報を基に、動作ルートの確認、動作時間の確認等を設計データと照視により比較検討し、問題点を見つけていた。このようなジャーナル情報取得方法の公知例としては、例えば、特開平4-133158号に開示された「オンライン・ジャーナル取得方式」などがある。

【0003】

【発明が解決しようとする課題】上記したように、従来はプログラムの動作過程を確認する方法として、プログラムに対しジャーナルポイントを人手により決め、決まったところにジャーナルを入れ、得られたジャーナル情報を基に動作ルートの確認・動作時間の確認を別に作成

している設計データと黙視により確認していた。しかしこのような方法では、システムが大規模化すると、ジャーナル情報も設計データも膨大なものになり、その中から必要なジャーナル情報を探し出し、それを基に設計データと比較確認するには莫大な時間を必要とし、また熟練者の感に頼らざるを得なくなる。このため、作業者の負担が増大し、また特に経験の少ない作業者にとっては、例えば「性能が十分にでない」原因や「故障」原因の解析に、どのような情報を取得すべきかの決定が難しかった。また、熟練者のジャーナル取得・検証のノウハウの引き継ぎは難しく、引き継がれても不十分な結果になっていることが多い。

【0004】本発明の目的は、プログラムが動作する計算機アーキテクチャに合わせ自動的にジャーナルを埋め込むことができ、また取得したジャーナル情報と設計データとの比較・検証が自動的に行え、さらに必要とする詳細なジャーナル情報を容易に取り出すことのできるジャーナル取得解析装置を提供することにある。

【0005】

【課題を解決するための手段】本発明は、解析対象プログラムを構成する処理単位のうち、どの処理単位がどの処理要因で起動されたときにその処理単位の始点あるいは始点と終点とにジャーナルポイントを設定するかをユーザが定義して格納しておくためのジャーナル定義格納手段と、この手段により定義されたプログラムのジャーナルポイントを検出してその位置へジャーナルを組み込むためのジャーナル組み込み手段と、前記処理単位毎に、その起動要因と起動要因対応の予想動作時間及び次に起動する起動処理単位とその起動要因をユーザが定義し格納するための動作条件定義格納手段と、解析対象プログラム実行時に前記ジャーナル組み込み手段により組み込まれたジャーナルにより出力されたジャーナル情報を取得し格納するためのジャーナル情報取得格納手段と、この手段により取得格納されたジャーナル情報と前記動作条件定義格納手段に格納された動作条件とをそれぞれ解析編集して比較し、その比較結果をユーザに表示するための比較表示手段と、を備えたことを特徴とするジャーナル取得解析装置を開示する。

【0006】更に本発明は、解析対象プログラムを構成する処理単位の始点と終点をジャーナルポイントとして検出し、その位置へジャーナルを組み込むためのジャーナル組み込み手段と、前記処理単位毎に、その起動要因と起動要因対応の予想動作時間及び次に起動する起動処理単位とその起動要因をユーザが定義し格納するための動作条件定義格納手段と、解析対象プログラム実行時に前記ジャーナル組み込み手段により組み込まれたジャーナルにより出力されたジャーナル情報を取得し格納するためのジャーナル情報取得格納手段と、この手段により取得格納されたジャーナル情報と前記動作条件定義格納手段に格納された動作条件とをそれぞれ解析編集して比

較し、その比較結果をユーザに表示するための比較手段と、を備えたことを特徴とするジャーナル取得解析装置を開示する。

【0007】更に本発明は、解析対象プログラムを構成する処理単位毎に、その起動要因と起動要因対応の予想動作時間及び次に起動する起動処理単位とその起動要因をユーザが定義し格納するための動作条件定義格納手段と、この手段により定義された動作条件からジャーナルポイントを自動生成してその位置へジャーナルを組み込むためのジャーナル組み込み手段と、解析対象プログラム実行時に前記ジャーナル組み込み手段により組み込まれたジャーナルにより出力されたジャーナル情報を取得し格納するためのジャーナル情報取得格納手段と、この手段により取得格納されたジャーナル情報と前記動作条件定義格納手段に格納された動作条件とをそれぞれ解析編集して比較し、その比較結果をユーザに表示するための比較表示手段と、を備えたことを特徴とするジャーナル取得解析装置を開示する。

【0008】更に本発明は、前記比較表示手段が、前記取得されたジャーナル情報と前記定義された動作条件とが一致しない処理単位と一致する処理単位とを区別して図示する機能と、処理単位を示す図形を指示手段で指示したときに当該処理単位に関する前記ジャーナル情報を表示する機能とを備えたことを特徴とするジャーナル取得解析装置を開示する。

【0009】更に本発明は、前記ジャーナル情報取得格納手段が、解析対象プログラムが複数台の計算機にネットワークを介して分割して搭載されている場合に、前記ジャーナル情報をネットワークを介して1つの計算機に集約して格納する集約手段を有していることを特徴とするジャーナル取得解析装置を開示する。

【0010】更に本発明は、解析対象プログラムに組み込むジャーナルを複数のグループに分割するとともに、前記ジャーナル情報取得手段が、前記ジャーナルのグループ毎に当該ジャーナルにより得られたジャーナル情報を分類して格納することを特徴とするジャーナル取得解析装置を開示する。

【0011】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照しながら詳細に説明する。図1は、本発明になるジャーナル取得解析装置の構成例を示すブロック図で、ジャーナル情報取得対象の計算機100、ジャーナル定義組み込み装置200、及びジャーナル取得解析装置300から成っている。計算機100には、ジャーナル情報取得対象の動作プログラム101、102、103、104、105、各プログラムの動作状況をジャーナルとして取得するジャーナル取得部106、取得したジャーナルを格納するジャーナル情報格納装置107が設けられている。ジャーナル定義組み込み装置200は、ジャーナル定義部201、計算機システムアーキテクチャ

に対応したジャーナル定義を格納するジャーナル定義格納装置202、計算機システムアーキテクチャに合わせて各プログラムにジャーナルを組み込むジャーナル組み込み部203からなっている。

【0012】ジャーナル取得解析装置300では、動作条件定義部301で自プログラムの起動要因、自プログラムの予想動作時間とその範囲、起動プログラム、起動プログラムの要因のプログラム予定動作条件を入力し、動作条件格納装置302に登録する。ジャーナル編集部306は、計算機100のジャーナル情報格納装置107からジャーナル情報を収集し、使用目的や種別毎に編集、分類をし、編集結果ジャーナル格納装置308に登録する。

【0013】動作比較検証部305は、動作条件格納装置302の登録情報を元に動作予想条件を解析する動作条件解析部304から予想動作情報を取得し、編集結果ジャーナル格納装置308からはプログラム実行結果のジャーナル情報を取得し、合致した条件のジャーナル情報を比較検証し、エラー情報を含めてプログラム動作表示部307に結果情報を渡す。プログラム動作表示部307は、その結果を元にプログラム動作の図式化を行い、端末401に表示する。図式化されたプログラムのうちエラーと情報を含んだものに関しては、色の変化・ブリンク・強調表示を行い、視覚的にエラーを目立たせる。また、端末400に表示されている図の一部を指定することにより、プログラム動作表示部307は、編集結果ジャーナル格納装置308から関連詳細ジャーナルを取り出し再び端末400に表示する。

【0014】動作条件作成部303は、編集結果ジャーナル格納装置308から前回の編集結果を取得し、これを別の動作条件の定義情報として動作条件格納装置302に登録する。動作条件定義部301は、動作条件格納装置302から動作条件を取り出し、端末400に表示し、確認後修正がある場合には再度新しい条件に編集し直し、動作条件格納装置302に登録する。これはエラーとなったプログラムを修正した場合に利用される。即ちその修正したプログラムを動作させて得たジャーナル情報をジャーナル情報格納装置107に登録し、これをジャーナル編集部306で編集して編集結果ジャーナル格納装置308に登録する。そして、この修正プログラムに関して取得したジャーナル情報と、前回の動作結果に基づいて生成した動作条件格納装置302に登録した動作条件とを比較すれば、プログラム修正の結果を確かめられる。むしろ設計データに基づく動作条件との比較も可能で、どの動作条件と比較するかは、作業者が端末から指示を与え、動作条件解析部304がその指示に従って対応する動作条件を用いることで決まる。ジャーナル定義変更部309は、動作条件格納装置302に定義してある動作条件を元にジャーナル定義情報をジャーナル定義格納装置202に登録する。

【0015】以上、発明のジャーナル取得解析装置の構成と大まかな動作の概要を説明したが、ジャーナル定義組み込み装置200及びジャーナル取得解析装置300は、計算機100内に設けてもよく、また別の計算機で実現してもよい。後者の場合はネットワークを経由して計算機100と接続してもよい。また、端末400、401は1つの端末であってもよいことは言うまでもない。更に、計算機100としては、リアルタイムOS、UNIX、WINDOWS等のどのアーキテクチャであってもよい。

【0016】次に、図1のジャーナル取得解析装置の動作を「従業員データベースから30歳の男性を探す」という具体例で説明する。図2は、リアルタイムOSにおけるプログラムA、B、C、D、E、Fの6プログラムの流れを示したものである。プログラムAはシステムにより自動的に起動され、図3(A)の画面を表示する。そしてプログラムAは、図3(A)の画面に表示された画面ボタンの「新規入力」・「検索」・「終了」を判断し、「新規入力」・「検索」のときはプログラムBを起動し、「終了」のときは後処理をして終了する。プログラムBは、画面に入力された入力項目に対し不正をチェックし、不正が無い場合は入力データを内容とするデータテーブルTGを作成し、そして「新規入力」の場合はプログラムCを、「検索」の場合はプログラムEを起動する。プログラムCは、データテーブルTGとデータベースDBHとの整合性をチェックして、データベースアクセスプログラムDを起動する。プログラムDは、データテーブルTGを基にデータベースDBHにデータを格納する。プログラムEは、データテーブルTGの内容がデータベースDBHにあるデータの範囲にあるかをチェックし、問題無しの場合プログラムDを起動する。プログラムDは、検索条件に従いデータベースDBHを検索して表示テーブルTIを作成する。プログラムFは、プログラムD終了後にプログラムEより起動され、検索結果として図3(B)のような画面を表示する。この例では、年齢30歳の男性が検索されており、この検索結果の表示にいたるまでのプログラム動作について、プログラム設計時のデータと実行結果との検証を行うものとする。

【0017】図4は、検証対象のリアルタイムOS対応のジャーナル定義格納装置202に格納されたジャーナル定義情報の例で、リアルタイムOSにおける起動と終了のイベントに関わるマクロ“queue”、“Stop”等とその組み込み条件が定義されており、各プログラムに対し、この定義されたマクロがあると、対応する組み込み条件に従ってジャーナル組み込み部203がジャーナルを該当するプログラムに組み込む。このように、ジャーナル定義情報をアーキテクチャ対応で定義しておくことで、ジャーナル組み込みが自動化できる。

【0018】一方、プログラム設計時に、設計者は動作

条件定義部301を使用し、動作条件格納装置302に動作条件を定義する。図5は、図2に示した従業員名簿作成検索プログラムの動作条件の定義であり、プロセスA～Fの各々に対して、それを起動する要因（「新」は「新規入力」の略）、起動されたときの予想動作時間、当該プロセスが次に起動する起動プロセスとそれを起動する要因が定められている。この内容は図2のプログラムの流れから作成できる。動作条件解析部304は、この格納装置302の定義を読み出して解析し、この結果はプログラム動作表示部307へ送られ、端末401に表示できるが、このときの表示画面は図6となる。

【0019】以上のように、対象プログラムにジャーナルが組み込まれ、また設計時の動作条件が定義された状態で、作業者が従業員データベースから30歳の男性を探すための操作を行うと、先ずプログラムAが起動され、図3（A）が表示される。ここで年齢欄に30、性別の男性を押し、検索ボタンを押すとプログラムが動作して図3（B）の画面が表示される。この時各プログラムに組み込まれているジャーナルをジャーナル取得部106が取得し、ジャーナル情報格納装置107に格納する。図7は、ジャーナル編集部306が、上記のようにしてジャーナル情報格納装置107に格納されたジャーナル情報を取り出し、編集した編集結果格納装置308のデータ例である。各プロセスのイベント発生時刻がジャーナル時間として記録されており、このジャーナル時間の差がそれぞれのイベントの動作に要した時間を表している。従ってこの実際の動作時間と、動作条件格納装置302に格納されている図5の定義データを動作条件解析部304で解析した結果とを、動作比較検証部305で比較すると、実動作を検証できる。その結果をプログラム動作表示部307で表示データとして表示したのが図8である。図8では、ジャーナル情報を比較解析した部分のみの表示となっており、プログラムB、E、Dそれぞれがエラーとして強調表示（図ではバッチ）されている。ここで“エラー”としたのは、図5のように定義した予想動作時間が、プログラムB（検索）、E、D（検索）に対してそれぞれ5000msec、4000msec、3500msecであったのに対し、取得した図7のジャーナル情報から求めた動作時間がそれぞれ7000msec、7200msec、6100msecといずれも大きくなっているからで、所望の性能が出ていない結果であったからである。そこで詳細ジャーナル情報の表示として、例えば図8のプログラムDをクリックすると、比較したジャーナル情報の詳細が表示され、データベースへのアクセスに予想外の時間がかかっているのがわかる。

【0020】図9は、図7のように得られ、編集結果ジャーナル格納装置308に格納された実績データをプログラム修正後の結果と比較するために、動作条件作成部303で図7のデータを編集し、動作条件定義部301を通して新しい動作条件として動作条件格納装置302

へ格納したものである。図9のプロセスBの予想動作時間が検索時に7400msecとなっているのは、図7のプロセスBの“Start”から同プロセスの“Stop”までの時間、図9のプロセスE検索時の予想動作時間が7200msecとなっているのは、図7のプロセスEの“Start”から“Stop”までの時間、図9のプロセスD検索時の予想動作時間が6100msecとなっているのは、図7のプロセスDの同じく“Start”から“Stop”までの時間である。また各データベース読み出し（Dbread）や表書き出し（Twrite）等の詳細な時間も付加されている。このような新しい“動作条件”を登録しておけば、プログラムを修正し再度動作させたときにその新しい動作条件と比較することで、今度は細部にわたり、比較検証が可能となる。

【0021】次に、図1のジャーナル取得解析装置の動作を、「電車の運行状態を検証する」場合について具体的に説明する。図10は、駅500から電車510がスタートして信号機520、踏み切り530を通過し、駅540に入るまでのジャーナル情報を示し、図11は、電車510、信号機520、踏み切り530に対して、動作条件定義部301を用いて、動作条件を定義し、動作条件格納装置に格納されたものである。図12は、この動作条件を元にプロセスの動作を図式化したプロセス構成図で、先の例の図6に相当する。図13は、電車が駅500からスタートして駅540に入るまでの間に、各プロセスから取得したジャーナル情報を、ジャーナル編集部306で編集した編集結果ジャーナル格納装置のデータで（先例の図7に対応）、図14は、このデータと動作条件格納装置の動作条件とを動作比較検証部305で比較検証した結果を図式化した比較検証プロセス構成図である。

【0022】図11の動作条件には、電車510、信号機520、踏み切り530それぞれの起動要因、予想動作時間、起動プロセス、起動プロセスの要因を定義してある。図13には、電車510発車時の電車スタートジャーナル情報、速度に関する電車速度ジャーナル情報、信号スイッチ動作の信号ジャーナル情報、信号機を電車が通過したときの信号通過ジャーナル情報、踏み切りスイッチの動作を示す踏み切りジャーナル情報、踏み切りを電車が通過したときの踏み切り通過ジャーナル情報、電車が駅に到達し停止したときの電車ストップジャーナル情報等が示されている。これらのジャーナル情報と動作条件とを比較検証した図14の比較検証プロセス構成図では、赤信号に変更する時間が、動作条件格納テーブルでは1secなのに対し3secかかっているため、エラーとして扱い強調表示（図ではハッチ）となっている。また、それぞれのプロセスにおける詳細なジャーナルがみたい場合、プロセスをキーにジャーナル情報を格納装置308から取り出し、即時に表示できる。ここでは電車の詳細ジャーナルを表示し確認している。また、先例と

同様に、図13のジャーナル情報を動作条件として用いれば、次に電車を起動させて出力したジャーナル情報と詳細部まで比較検証できるため、より細かな動作チェックが可能である。

【0023】次に、図1のジャーナル取得解析装置の動作を、「電車の駅・運行時間帯に変動する乗客調査を行う」場合について具体的に説明する。図15は、駅600から乗客をのせ駅640・駅650で停車し、最終駅660に至るまでに取得するジャーナル情報を示しており、図16は、電車610、電車のドア620が、それぞれの駅を経由して最終駅に至るまでの動作条件を定義し、動作条件格納装置302に格納した動作条件である。図17は、この動作条件を図式化して示したプロセス構成図である。図18は、電車が駅600をスタートしてから各駅を経由して最終駅660に入るまでに、各プロセスから取得したジャーナル情報を編集し、編集結果ジャーナル格納装置へ格納したデータ、図19は、このジャーナル情報と図16に定義した動作条件とを比較検証して図式化した比較検証プロセス構成図である。

【0024】図16の動作条件には、図15における電車610、ドア620及びドアセンサ630に対し、各駅で起こりうるそれぞれの起動要因、予想動作時間、起動プロセス、起動プロセスの要因を定義してある。図18には、電車の各駅での発車を示す電車スタートジャーナル情報、電車の到達停止を示す電車ストップジャーナル情報、各駅でのドア開閉を示すドア開閉ジャーナル情報、各ドアに取り付けられたドアセンサによる乗客感知のドアセンサジャーナル情報等が示されている。これらのジャーナル情報と動作条件とを比較検証した図19の比較検証プロセス構成図では、駅650におけるドアの開閉に1分30秒かかっており、図16に定義した予定の1分±10秒からかけ離れているため、エラーとして扱い強調表示（ハッチ）となっている。また、それぞれのプロセスにおける詳細なジャーナルがみたい場合、プロセスをキーにジャーナル情報を格納装置308から取り出し、即時に表示できる。ここではエラー表示となっている駅3：電車プロセスと駅3電車ドアプロセスの詳細ジャーナルを表示し確認している。この詳細ジャーナルを見る限りにおいては、ドアセンサの反応が終了しないうえにドアを開められなかったことが読み取れる。即ち乗降客が多いための結果であると判断できる。また、先例と同様に図18のジャーナル情報を動作条件として用いれば、次に電車を起動させて出力したジャーナル情報と詳細部まで比較検証できるため、より細かな動作チェックが可能である。

【0025】以上に示したような本発明のジャーナル取得解析装置を用いれば、例えば同じ曜日の同じ時間帯を複数回検証することにより、時間帯による各駅の停車時間の変更調査がスムーズに行えるようになり、刻々と変化する乗客数に対する列車運行の時間制御をより正確に

行えるようになる。また、帰省シーズン・行楽シーズンの過去の解析結果を元に、電車の運行状況が事前に予測可能となり、スムーズな列車運行管理が行えるようになる。

【0026】なお、以上の説明では、ジャーナル定義組み込み装置200において、ユーザがジャーナル定義部201によりジャーナル定義を行ってこれを予めその格納装置202に格納しておくものとしたが、部分的あるいは全面的にユーザによるジャーナル定義を省く方法も可能である。その1つは、プログラムの適当なまとまり、例えばプロセス毎、ジョブ毎といったプログラム単位毎に、その始めと終わりはOSにより認識されているから、OSがそうした始まりと終わりを取り出しそこへ該当するジャーナルを自動的に設定するものである。

【0027】また、別の方法は、定義されて動作条件格納装置302に格納された動作条件から自動的にジャーナルポイントを決めるものである。この具体的な方法としては、動作条件格納装置302に格納された動作条件の内容をジャーナル定義変更部309で解析し、ジャーナル定義格納装置202に格納することである。図5に示してある従業員名簿作成プログラムの場合には、その動作条件を元に、ジャーナル定義変更部309では、プログラムAからプログラムBは「新・検索」という条件で起動され、プログラムBからプログラムCは「新・プログラムDは「検索」という条件でそれぞれ起動されることを認識する。そしてこの認識に基づいて起動の条件で“queue”が自動作成され、起動するからには停止することをふまえ、停止条件である“stop”、“abort”が自動生成される。組み込み条件には無条件に「動作前」が定義され、図4のようなジャーナル定義格納テーブルがジャーナル定義格納装置202に作成される。これを元にジャーナル定義部203が各プログラムにジャーナルを組み込む。

【0028】

【発明の効果】本発明によれば、予めプログラムの起動要因・動作順序・動作時間を定義しておき、これと取得したジャーナル情報とを自動的に比較・解析してビジュアルに表示できる。従って種々のプログラム動作・性能に関わるトラブル情報の確認が容易となり、トラブル対策が効率的に行える。また、どのようなジャーナル情報を取得するかの設定も容易になる。さらに、取得したジャーナル情報を動作条件とみなしてプログラム変更後等に取得したジャーナル情報と比較することで、プログラム変更等の変化の影響を容易に確認できる。

【図面の簡単な説明】

【図1】本発明になるジャーナル取得解析装置の構成例を示すブロック図である。

【図2】従業員名簿作成・検索プログラムの例を示す流れ図である。

【図3】図2のプログラム動作時に表示される画面の例

である。

【図4】リアルタイムOSでの計算機システムアーキテクチャに対応したジャーナル定義定義格納テーブルの説明図である。

【図5】従業員名簿作成検索プログラムの動作条件を定義した図である。

【図6】図5の定義を表示したプログラム構成図である。

【図7】図2のプログラム実行時に取得したジャーナル情報の例である。

【図8】図7の結果と動作条件を比較検証した結果を示す図である。

【図9】図7の結果を動作条件としての定義した図である。

【図10】電車運行状態を示す図である。

【図11】図10の運行に関する動作条件を定義した図である。

【図12】図11の定義を表示したプロセス構成図である。

【図13】図10の運行時に取得したジャーナル情報の例である。

【図14】図13の結果と動作条件を比較検証した結果を示す図である。

【図15】電車運行状態の別の例を示す図である。

【図16】図15の運行に関する動作条件を定義した図である。

【図17】図16の定義を表示したプロセス構成図である。

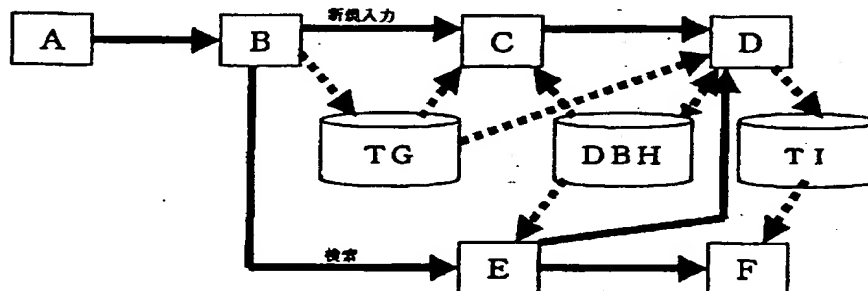
【図18】図15の運行時に取得したジャーナル情報の例である。

【図19】図18の結果と動作条件を比較検証した結果を示す図である。

【符号の説明】

100 計算機
101～105 プログラム
106 ジャーナル取得部
107 ジャーナル情報格納装置
200 ジャーナル定義組み込み装置
201 ジャーナル定義部
202 ジャーナル定義格納装置
203 ジャーナル組み込み部
300 ジャーナル取得解析装置
301 動作条件定義部
302 動作条件格納装置
303 動作条件作成部
304 動作条件解析部
305 動作比較検証部
306 ジャーナル編集部
307 プログラム動作表示部
308 編集結果ジャーナル格納装置
309 ジャーナル定義変更部
400、401 端末

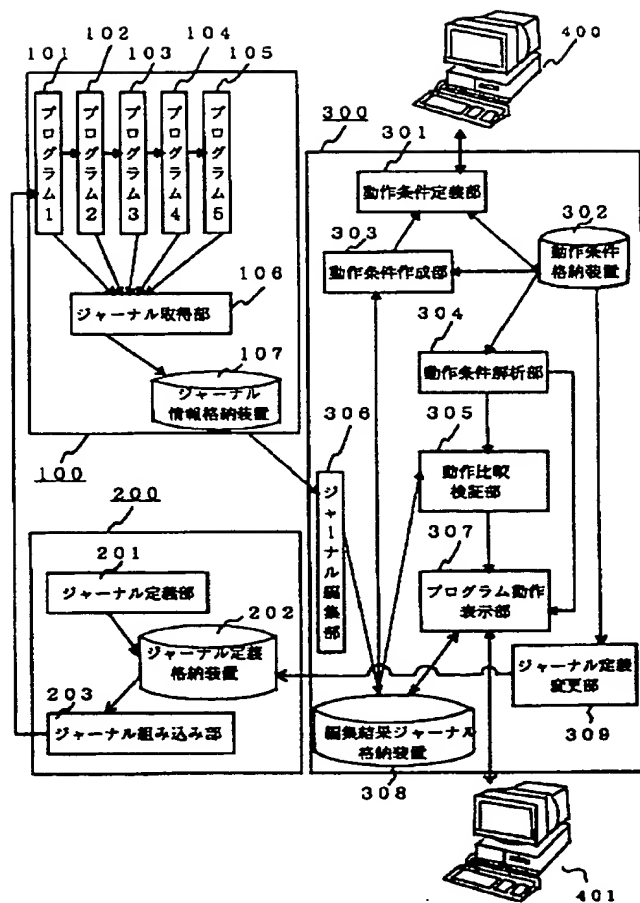
【図2】



【図4】

ジャーナル組み込み対象箇所	組み込み条件
Queue	動作前
Stop	動作前
Abort	動作前
Release	動作前
Wait	動作前
Main	関数の先頭と終わり

【図1】



【図7】

プロセス	イベント	要因	ジャーナル時間
A	Start		98.2.20 10:13:28:000
B	Start	検索	98.2.20 10:13:40:000
E	Start	検索	98.2.20 10:13:40:100
E	Obread		98.2.20 10:13:40:300
E	Obread		98.2.20 10:13:40:500
D	Start	検索	98.2.20 10:13:40:600
D	Obread		98.2.20 10:13:40:700
D	Obread		98.2.20 10:13:41:400
D	Obread		98.2.20 10:13:42:100
D	Obread		98.2.20 10:13:42:800
D	Obread		98.2.20 10:13:43:500
D	Obread		98.2.20 10:13:44:200
D	Obread		98.2.20 10:13:44:900
D	Obread		98.2.20 10:13:45:600
D	Obread		98.2.20 10:13:46:300
D	Write		98.2.20 10:13:48:600
D	Stop		98.2.20 10:13:48:700
F	Start		98.2.20 10:13:48:750
F	Disc		98.2.20 10:13:47:050
F	Stop		98.2.20 10:13:47:200
E	Stop		98.2.20 10:13:47:300
B	Stop		98.2.20 10:13:47:400
A	Stop		98.2.20 10:15:57:100

【図3】

(A)

従業員名簿

氏名

年齢

性別 ☐ 男性 ☐ 女性

従業員コード

新規入力 検索 終了

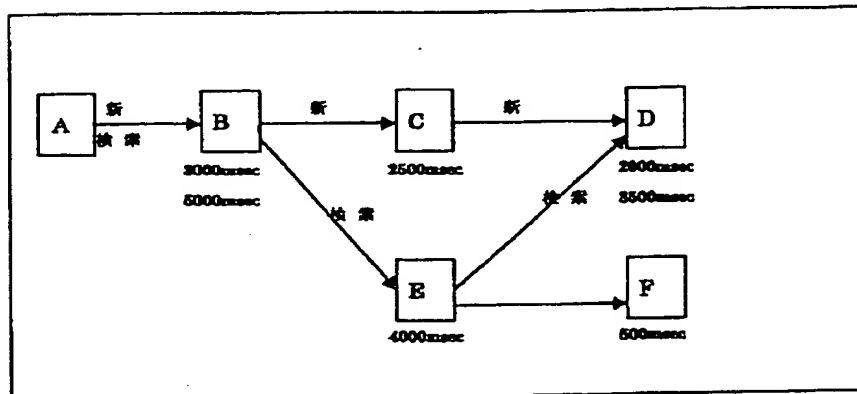
(B)

氏名	年齢	性別	従業員コード
山田太郎	30	男性	00305
山田次郎	30	男性	00306
山田三郎	30	男性	00307
佐藤一男	30	男性	00308
佐藤次男	30	男性	00309
佐藤三男	30	男性	00310
高橋一	30	男性	00312

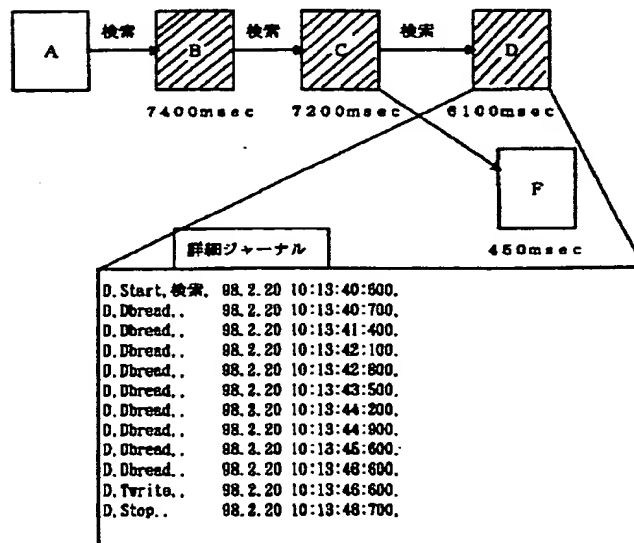
【図 5】

プロセス	起動要因	予起動作時間	起動プロセス	起動プログラムの要因
A	—	—	B	新
B	新	3000± 300msec	B	検索
	検索	5000± 500msec	C	新
C	新	2500± 300msec	E	検索
E	検索	4000± 400msec	D	新
D	新	2000± 300msec	D	検索
	検索	3500± 400msec		
F	—	500± 100msec		

【図 6】



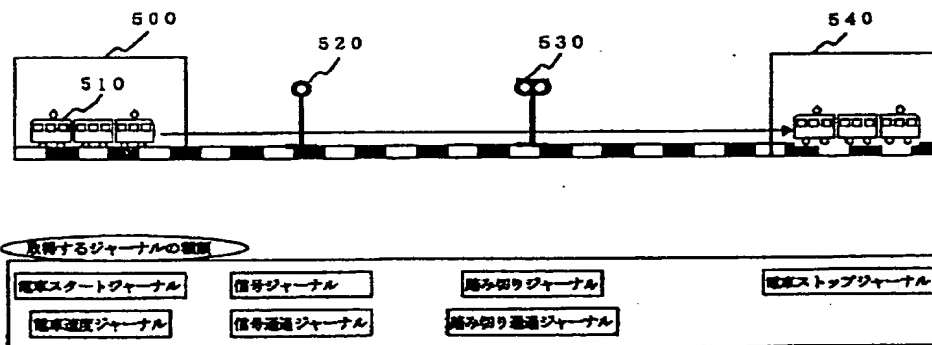
【図 8】



【図9】

プロセス	起動要因	イベント	予推動作時間	起動プロセス	起動プログラムの要因
A	—		—	B	新
				B	検索
B	新		3000±800msec	C	新
	検索		7400±500msec	E	検索
C	新		2500±800msec	D	新
E	検索		7200±400msec	D	検索
E		Dbread	200msec		
E		Dbread	100msec		
D	新		3000±300msec		
	検索		6100±400msec		
D		Dbread	700msec		
D		Dbread	700msec		
D		Dbread	700msec		
D		Dbread	700msec		
D		Dbread	700msec		
D		Dbread	700msec		
D		Dbread	700msec		
D		Dbread	700msec		
D		Twrite	100msec		
F	—		450±100msec		
F		Disp	150msec		

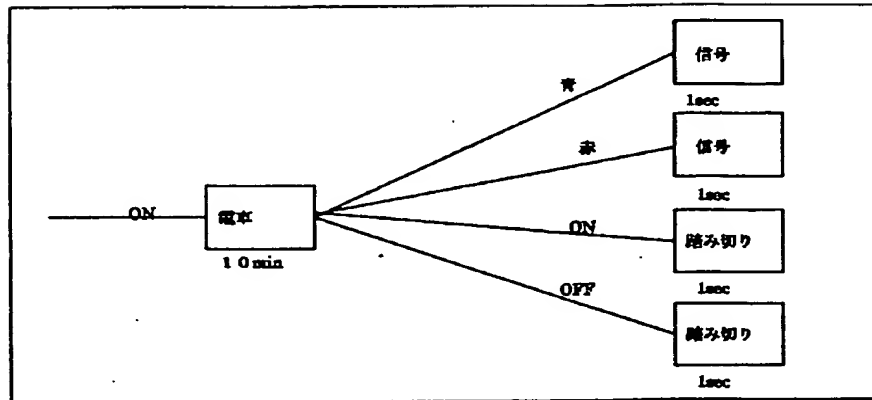
【図10】



【図11】

プロセス	起動要因	予推動作時間	起動プロセス	起動プログラムの要因
電車	ON	10min	信号	青
			信号	赤
			踏み切り	ON
			踏み切り	OFF
信号	青	1sec		
信号	赤	1sec		
踏み切り	ON	1sec		
踏み切り	OFF	1sec		

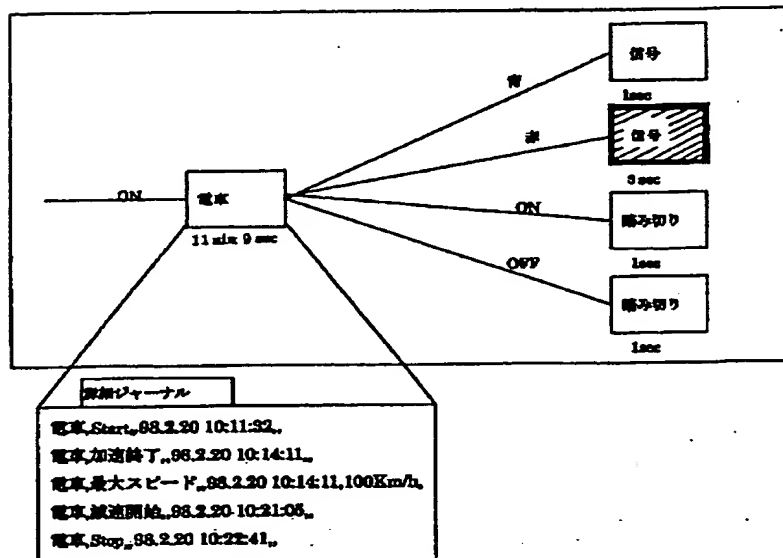
【図12】



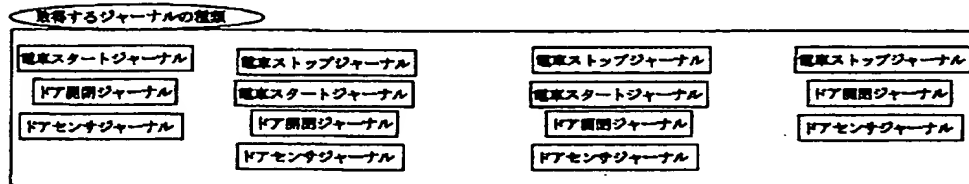
【図13】

プロセス	イベント	要因	ジャーナル時間	その他
電車	Start		98.2.20 10:11:32	
信号	Start	青	98.2.20 10:13:28	
信号	Stop		98.2.20 10:13:29	
電車	加速終了		98.2.20 10:14:11	
電車	最大スピード		98.2.20 10:14:11	100Km/h
信号	先頭通過		98.2.20 10:14:58	
信号	最後尾通過		98.2.20 10:18:08	
信号	Start	赤	98.2.20 10:18:52	
信号	Stop		98.2.20 10:18:55	
踏み切り	Start	ON	98.2.20 10:17:12	
踏み切り	Stop		98.2.20 10:17:13	
踏み切り	先頭通過		98.2.20 10:18:52	
踏み切り	最後尾通過		98.2.20 10:19:13	
踏み切り	Start	OFF	98.2.20 10:19:17	
踏み切り	Stop		98.2.20 10:19:18	
電車	減速開始		98.2.20 10:21:05	
電車	Stop		98.2.20 10:22:41	

【図14】



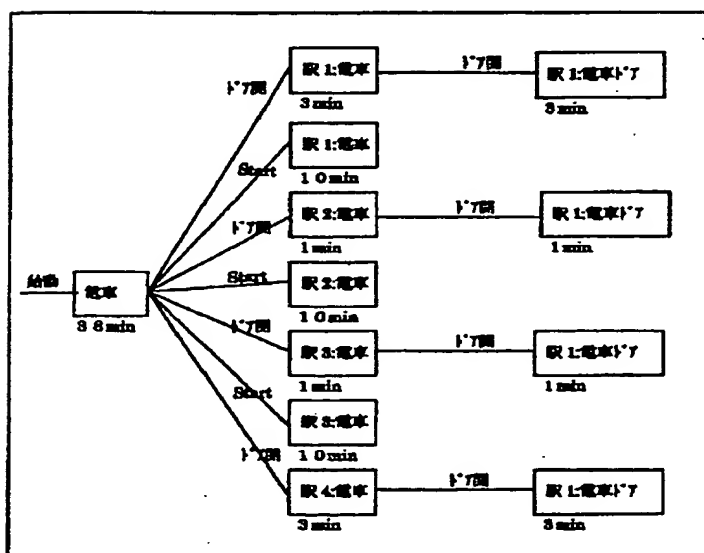
【図15】



【図16】

プロセス	起動要因	予想動作時間	起動プロセス	起動プログラムの要因
電車	始動	38min+2min	駅1:電車	ドア開
			駅1:電車	Start
			駅2:電車	ドア開
			駅2:電車	Start
			駅3:電車	ドア開
			駅3:電車	Start
			駅4:電車	ドア開
駅1:電車	ドア開	3min+30sec	駅1:電車ドア	ドア開
駅1:電車ドア	ドア開	3min+30sec		
駅1:電車	Start	10min+1min		
駅2:電車	ドア開	1min+10sec	駅2:電車ドア	ドア開
駅2:電車ドア	ドア開	1min+10sec		
駅2:電車	Start	10min+1min		
駅3:電車	ドア開	1min+10sec	駅3:電車ドア	ドア開
駅3:電車ドア	ドア開	1min+10sec		
駅3:電車	Start	10min+1min		
駅4:電車	ドア開	3min+30sec	駅1:電車ドア	ドア開
駅4:電車ドア	ドア開	3min+30sec		

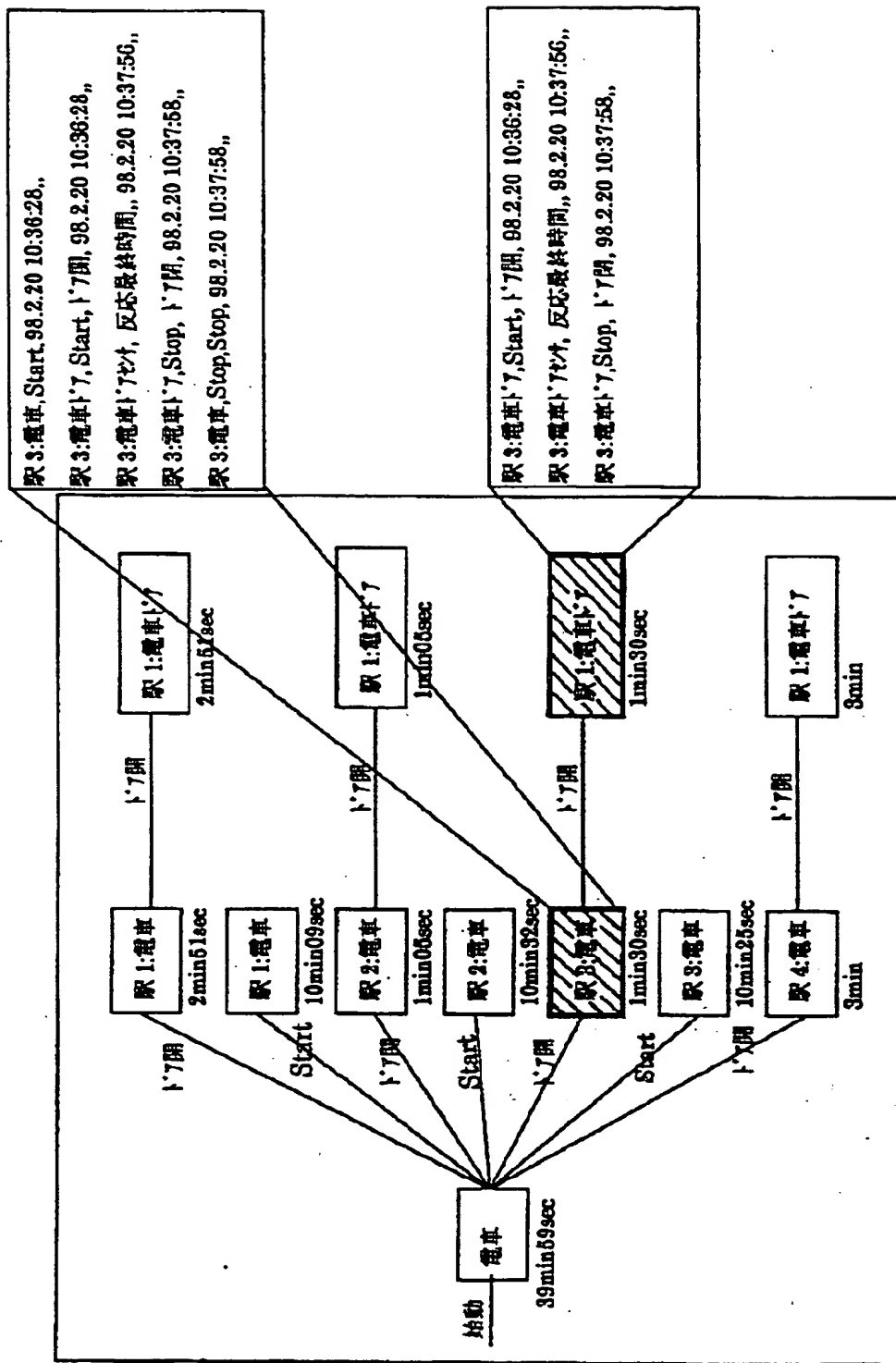
【図17】



【図18】

プロセス	イベント	要因	ジャーナル時間	その他
電車	始動		98.2.20 10:11:32	
駅1:電車	Start	ドア開	98.2.20 10:11:38	
駅1:電車ドア	Start	ドア開	98.2.20 10:11:38	
駅1:電車ドアセンサ	反応最終時間		98.2.20 10:13:28	
駅1:電車ドア	Stop	ドア閉	98.2.20 10:14:29	
駅1:電車	Stop	Stop	98.2.20 10:14:29	
駅1:電車	Start	Start	98.2.20 10:14:33	
駅1:電車	Stop	Stop	98.2.20 10:24:42	
駅2:電車	Start	ドア開	98.2.20 10:24:43	
駅2:電車ドア	Start	ドア開	98.2.20 10:24:43	
駅2:電車ドアセンサ	反応最終時間		98.2.20 10:25:28	
駅2:電車ドア	Stop	ドア閉	98.2.20 10:25:49	
駅2:電車	Stop	Stop	98.2.20 10:25:49	
駅2:電車	Start	Start	98.2.20 10:25:53	
駅2:電車	Stop	Stop	98.2.20 10:36:25	
駅3:電車	Start	ドア開	98.2.20 10:36:28	
駅3:電車ドア	Start	ドア開	98.2.20 10:36:28	
駅3:電車ドアセンサ	反応最終時間		98.2.20 10:37:56	
駅3:電車ドア	Stop	ドア閉	98.2.20 10:37:58	
駅3:電車	Stop	Stop	98.2.20 10:37:58	
駅3:電車	Start	Start	98.2.20 10:38:00	
駅3:電車	Stop	Stop	98.2.20 10:48:25	
駅4:電車	Start	ドア開	98.2.20 10:48:30	
駅4:電車ドア	Start	ドア開	98.2.20 10:48:30	
駅4:電車ドアセンサ	反応最終時間		98.2.20 10:50:40	
駅4:電車ドア	Stop	ドア閉	98.2.20 10:51:30	
駅4:電車	Stop	Stop	98.2.20 10:51:30	
電車	Stop		98.2.20 10:51:31	

【図19】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.